Language Modelling for Spoken Dialogue Systems; Grammar-Based and Robust Approaches Compared and Contrasted

Genevieve Gorrell

December 22, 2003

1 Introduction

As spoken dialogue systems increase in their capabilities, becoming able to support more complex and diverse domains, the pressure is on to provide speech recognition capabilities appropriate to their demands. High performance speech recognition depends on high-quality language modelling.

Current work in language modelling focuses on two main areas; formal and stochastic approaches. Formal approaches to language-modelling come in many forms and serve many motivations. Simply put, formal approaches are about hand-coding definitions of a language. A variety of formalisms exist for doing this, mostly derived from Chomsky's formal language theory. Beyond this, little can be assumed. Uses range from linguistically-motivated attempts to describe a natural language through to specifications of allowable phrases in limited-domain voice recognition systems. Stochastic approaches, on the other hand, involve compilation of a finite-state machine in which the likelihood of a given word occurring is calculated based on the corpus, possibly given the context of the preceding n words; two is common.

Each approach has advantages and disadvantages with regards to use in the speech-recognition component of a spoken dialogue system. Development time, reusability and expertise required to create the language model all play a role in determining the appropriate solution in many cases. Furthermore, the nature of the system affects which solution will perform best; range of language required to be covered and the experience of the users critically affect how functional each solution can be.

In the following two sections each of the two main paradigms is introduced. Performance strengths and weaknesses are discussed. Section 4 draws some comparisons between the two approaches before concluding.

2 Grammar-Based Language Modelling

The thriving research field of formal language theory finds itself relevant far beyond the study of natural languages. Artificial languages such as programming languages can be classified according to the Chomsky Hierarchy, which describes a hierarchy of formalisms for describing a language, each of which has different limitations;

- Type 0, the general rewriting system, in which any symbol or sequence of symbols can be rewritten unrestrictedly to any other
- Type 1, the context-sensitive rewriting system, in which a single nonterminal symbol can be replaced with another symbol or sequence of symbols given a particular context. Unification grammars fall into this category.
- Type 2, the context-free rewriting system, in which any single nonterminal symbol can be replaced with another symbol or sequence of symbols regardless of context.
- Type 3, the finite state, or regular, system, in which any single nonterminal can be rewritten to a terminal, or a terminal followed by a nonterminal (possibly itself), or the empty string.

The differences between these formalisms is perhaps more intuitively explained with reference to what they *cannot* describe. A type 3 language could for example describe a grammar in which the first word can appear one or more times, followed by the second word one or more times. A type 2 language can describe that too, but unlike the type 3, can describe a grammar in which the first word can appear one or more times, and then the second word appears the same number of times as the first. This could be brought about using a grammar rule set such as;

```
A -> []
A -> aAb
```

It is apparent that such a rule is allowable in a context-free grammar but not a regular one. Likewise, however, a language in which the first word can appear one or more times, followed by the second word the same number of times, followed by the third word the same number of times can be described using a type 1 grammar, but not a type 2. The rule set;

```
A -> aABC       A -> aBC
aB -> ab        C -> c
bB -> bb        CB -> BC
```

allows derivations such as the following;

```
A -> aABC -> aaBCBC -> aabCBC -> aabBCC -> aabbCC -> aabbcC -> aabbcc
```

It is evident that the above rule set could not be encoded in a context-free formalism. The context-sensitivity is required to specify the difference in behaviour between B in bB and CB for example.

The most common grammar formalism used in language modelling for speech recognition is the type 2 context-free formalism. Unification grammars, of type 1, are also in consideration for the task. For example, [**?**] demonstrate that a unification grammar, a richer formalism than the context-free, can be compiled into an approximated context-free grammar for use in speech recognition.

In spoken dialogue systems, grammars are restricted in coverage to the domain of the system. They are often coded specifically for the application, or even a particular dialogue state within the application. Since grammars are a hand-coded description of the language, they tend to be quite restrictive; the grammar may for example encode only grammatical English, precluding the user speaking ungrammatically, or changing their mind mid-sentence. The grammar-writer may not have anticipated how the user will phrase themselves. However, an utterance recognised with a grammar is an utterance that is automatically parsed. A grammar might be annotated with semantics, such that an utterance recognised using it is also understood. Here is an example of a small grammar, annotated with semantics, written in the Nuance Grammar Specification Language [3];

```
MyGrammar
  (Greet:d world) {<hello_or_goodbye \$d>}

Greet
  [
     hello {return (hello)}
     goodbye {return (goodbye)}
  ]
```

It is a context-free grammar. It allows the sentences "hello world" and "goodbye world". Words in lower case are words that you can say. Upper-case words are grammar names. Semantic annotation is in curly brackets. Round brackets mean that the words are said in sequence ("and"); square, that only one of them is said ("or").

Grammar-based speech recognition is widely used in live commercial spoken dialogue systems, for example, [7]. Furthermore, several experimental systems have chosen this approach, for example, the On Off House system [1]. Pragmatic reasons for preferring the grammar-based approach include, for example, that the collection of a sufficiently extensive corpus to allow for the creation of a high-performance statistical language model (SLM) is time-consuming and expensive, making grammars an attractive option. Furthermore, grammars can be manipulated online, for example, to include new vocabulary, where a stochastic language model would have to be recompiled from an appropriate corpus. Where systems need to be created quickly to cover diverse domains and specific coverage, grammar-based language modelling has the edge. It is also

very appropriate in a system where users can be expected to become experts in using the system, as in for example command and control systems, such as CommandTalk [2]; naturalness of language use can be sacrificed for fast and accurate recognition.

In terms of actual performance, a good stochastic language model is hard to beat. However, if users keep themselves within coverage of the grammar (as an experienced user can be expected to) then the grammar can have the edge. Knight et al. [1] compared two versions of the same system over the same corpus, the first version using grammar-based recognition and the other, stochastic. They showed that the grammar-based system outperformed the SLM where the user spoke within coverage of the grammar, both on word error rate, and very markedly on semantic error rate (where the grammar's parsing side-effect served to further reduce error rate.)

3 Stochastic Language Modelling

Stochastic (or N-gram) language modelling is a line of research that grew out of the need to reduce the search space in a whole range of natural language problems where a word stream is subject to noise and the original stream needs to be recovered. Speech recognition clearly falls into this category. The N-gram encodes the probability of a word occurring given the preceding n words as context.

Begin by supposing that a natural language can be characterised as a series of conditional probabilities, in which;

$$P(w_1^k) = P(w_1)P(w_2|w_1)...P(w_k|w_1^{k-1})$$

That is to say, the probability of a word string is the probability that each word occurred given the preceding context. In terms of predicting the next word, w_1^{k-1} is the history and w_k is the prediction. An N-gram language model approximates this by supposing that two histories are equivalent if they end in the same n-1 words. A basic 3-gram language model encodes, therefore, for every word, the probability of that word occurring given the predecing two words;

$$P(w_n|w_1^{n-1})$$

Since word frequency in natural languages is Zipfian, that is to say that most of the words occur very rarely indeed, no corpus will cover all the word sequences a language model is likely to encounter in use. This means that many phrases that the language model encounters it has no probability for at all. Consequently, some form of smoothing is required. If our language model was a 3-gram, where a sequence of 3 words was unknown to the language model, the 2-gram could be used instead. Likewise if that was unknown, the unigram could be used. This "back-off" smoothing is one of a variety of strategies in common usage.

Another way to maximise corpus usage is to group words into classes. Sup-

pose your corpus contained two words that typically appeared in very similar contexts, for example, "november" and "february". The corpus contains examples of contexts for "february" that might be supposed to be equally applicable to "november". This information could be included in the language model by grouping those words together for the sake of assigning the probabilities. The resulting language model would be a "Class N-Gram", where "class" refers to the groups of words.

A final variant falls between formal and stochastic approaches. A probabilistic context-free grammar is a context-free grammar that has been augmented with probabilities, usually derived from a corpus. Such probabilities are important in handling ambiguity, and produce performance improvements over unaugmented context-free grammars.

Stochastic language models are the preferred choice in many experimental systems, where maximal permissiveness is important and a sufficiently large corpus is available. An SLM can be more flexible in the range of language and constructions it allows, since only n words are considered, unlike the grammar, which requires that the entire utterance is covered. Being entirely corpus-based, the SLM can model more accurately the language it can expect to receive as input, where the coverage of the grammar is more at the whim of the developer. That assumes, of course, that the corpus is appropriate to the system. Where the majority of users are novices, as in for example information systems such as JUPITER [6], the permissiveness of the language model becomes very important. Also, in an experimental system, where the domain can be matched to the corpus available and the emphasis is on the interpretation and dialogue management lying behind the recognition, the stochastic approach becomes very attractive.

4 Comparison and Discussion

Grammar-based and stochastic language modelling clearly have different advantages. In terms of actual recognition performance, their behaviour also shows marked differences. As mentioned previously, Knight et al. [1] compared recognition performance in a statistical and a grammar-based speech recogniser within the same domain. They showed that where users spoke in coverage of the grammar, the grammar-based recogniser had a small performance edge on both word error rate and on producing a correct semantic interpretation. But where the user spoke out-of-coverage, the performance of the grammar-based recogniser dropped off much more quickly than the performance of the SLM. Where users do not know what they can say to the system, for example, because they are new to the system, we can expect a stochastic language model to perform better than a grammar. We can also expect that as input starts to diverge from the coverage of the language model, the stochastic performance will degrade more gracefully than the grammar-based.

To summarise, formal methods cover a spectrum, through from linguistically-motivated descriptions of the language to adhoc recognition grammars for lim-

ited domain spoken dialogue systems. Advantages to the formal approach for speech recognition include that grammars encode deeper linguistic knowledge which can be made use of later; even the non-linguistically motivated ones remove the need for parsing. Practical advantages include the reduced need for a corpus and increased runtime and development-time flexibility. Furthermore, in systems where users can be made to learn the grammar, the grammar can them be written such as to maximise recognition performance, for example, by phrasing allowable utterances such that they do not sound similar. Stochastic approaches on the other hand win out on performance where users use varied language and do not become experts in using the system. Furthermore, since the knowledge about the language encoded in the model comes straight from the corpus, creating the model is a less skilled job.

Interesting directions currently being taken in the field include Explanation-Based Learning [4], where a large, linguistically-motivated reference grammar is specialised for use in limited-domain systems using a corpus of examples. Adaptivity in language models is a further topic of interest, where use is made of, for example, the fact that words that have recently occurred are more likely to reoccur to adjust the probabilities.

References

[1] S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. Comparing grammar-based and robust approaches to speech understanding: a case study. In *Proceedings of Eurospeech 2001*, pages 1779–1782, Aalborg, Denmark, 2001.

[2] R. Moore, J. Dowding, H. Bratt, J. Gawron, Y. Gorfu, and A. Cheyer. CommandTalk: A spoken-language interface for battlefield simulations. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 1–7, 1997.

[3] Nuance. http://www.nuance.com, 2003. as of 22 December 2003.

[4] M. Rayner. Applying explanation-based generalization to natural-language processing. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 1267–1274, Kyoto, Japan, 1988.

[5] M. Rayner, B.A. Hockey, and F. James. Compiling language models from a linguistically motivated unification grammar. In *Proceedings of the Eighteenth International Conference on Computational Linguistics*, 2000.

[6] S. Seneff, E. Hurley, C. Pao, P. Schmid, and V. Zue. Galaxy-II: A reference architecture for conversational system development. In *Proceedings of the 5th International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[7] TellMe. http://www.tellme.com, 2001. as of 15 March 2002.